Sometimes, Less Is More: Why Training More Simpler Models May Be Better

Jerry Liu   Yuke Wu

Computer Science, Stanford

Problem & Background

The goal of this project is to create a question answering model that works well on SQuAD 2.0 which contains both difficult no-answer questions and traditional answerable contextual questions. BiDAF [1] used to be state-of-the-art, but as a RNN-based model, it is difficult to parallelize and has trouble capturing long-term dependencies and contexts. We adopted several techniques from QANet [3] and implemented the full QANet architecture to see if transformer-based models have an architectural advantage over BiDAF. We also investigated how much model ensembling can help improve performance.

Methods

Improving BiDAF

- **Character-level Embedding**. We added character-level embedding with both $d = 64$ and $d = 200$ and concatenated it with word embedding.
- **Character-level CNN**. We experimented with a 2D-CNN on top of character embedding with 100 filters of kernel size $(1, 5)$ as specified by the BiDAF paper. This output goes through MaxPool of size $(1, 16)$ where 16 is the maximum number of characters per word.
- **Self-attention**. We also implemented a single self-attention layer based on the original transformer paper [2]. The self-attention output is concatenated with the BiDAF Context2Query and Query2Context attentions, shown in Figure 1.

Implementing QANet

We implemented QANet [3] from the ground up. The architecture consists of five major layers: input embedding, embedding encoder, context-query attention, model encoder, and output. We reused the BiDAF context-query attention layer due to its similarity to QANet attention.

- **1D Convolutions**. We replaced the Linear projection matrices with 1D Convolutions of kernel size 1 described in the QANet paper [3].
- **Character-level CNN**. We experimented with a 2D-CNN similar to the one used in BiDAF except with 128 filters.
- **Stochastic Depth**. We implemented stochastic depth (layer dropout) within the embedding and model encoder layers.

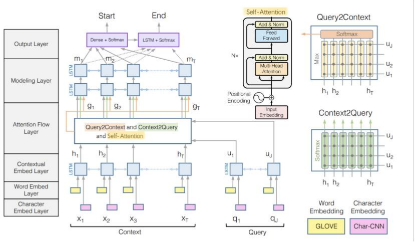

Figure 1. BiDAF with Self-Attention Layer

Methods (cont.)

Model Ensemble

- **Maximum Pair Probability**. For a given model $m_i, i = 1, \dots, n$, $p_{s_i}, p_{e_i} \in \mathbb{R}^l$ are the probability vectors for the start and the end positions where l is the length of the context.
  We constructed the probability matrix for all pairs of positions as $P_i = p_{s_i} p_{e_i}^\top$.
  Let $P_{i_{j,k}}$ be the (j, k) entry of matrix P_i, we define
  $$P_{j,k} = \max_{1 \le i \le n} P_{i_{j,k}}$$
  Then, we pick the start and end position s, e to be
  $$s, e = \underset{0 \le i < j < l}{\operatorname{argmax}} P_{i,j}$$
  This ensures the selection of the most confident answer (the pair with the highest joint probability) from all models. Note $s = e = 0$ indicates no-answer.
- **Automatic Mixed Precision**. To accelerate training, we integrated a CUDA feature – *Automatic Mixed Precision* which allows the Tesla V100 GPU to fully utilize its Tensor Cores' FP16 performance. We saw a 18% - 30% reduction in training time after implementing AMP. This makes the ensemble models more plausible.

Results

Model	Char Emb Dim	Variant	Exact Match	F1	AvNA
BiDAF	None	Baseline	57.70	61.25	68.02
	64	Baseline, Char Emb	60.71	64.26	70.63
	64	Self-Attention, Char Emb	61.17	64.38	70.51
	200	Self-Attention, Char Emb	**64.80**	68.00	73.75
	200	Self-Attention, Char Emb, Char CNN	63.01	66.07	71.80
QANet	64	Baseline	62.53	66.52	73.28
	200	Baseline	64.02	**68.14**	**74.79**
	200	Stochastic Depth	64.43	68.00	74.02
	200	Char CNN	63.75	67.28	73.53
Ensemble-5	200	3 BiDAF + 2 QANet	69.05	71.78	76.63
Ensemble-7	200	4 BiDAF + 3 QANet	**69.47**	**71.96**	**76.64**

Table 1. Performance on the SQuAD 2.0 dev set

Conclusions

- We found that **BiDAF with self-attention reached a similar performance** as vanilla QANet while taking **much less time to train**. QANet had similar performances across all question types. BiDAF performed particularly poorly in the "why" questions, possibly a result of the inherent disadvantage of RNNs to capture long-term depdendencies.
- Utilizing CUDA features such as AMP **significantly reduces** a model's training time.
- *Character-level CNN*, *varying learning rate* and *stochastic depth* **did not improve performance** on either BiDAF or QANet. Ensembling **greatly improved performance**, reaching EM/F1 score of 69.47/71.96, currently **top 3** on the dev set leaderboard.
- **Model ensembling**, meant to reduce neural network's variance, showed that **combining simpler, easier to train** models may provide a much **bigger gain** in terms of performance given the same amount of training time.
- An F1/EM score of $\sim 68/64.5$ may be the **limit** of the Input Embedding layer without more complex word or character embeddings. Future work should focus on whether a more complex Char-CNN or other encoder could improve performance.

Analysis

- Self-Attention with character embedding ($d = 200$) showed a remarkable performance uplift over the baseline BiDAF model and its EM/F1 scores are on par with vanilla QANet.
- QANet learned much faster in the first few epochs. Then, BiDAF continued to learn from (and overfit) the training data while QANet's training NLL plateaued. This is shown in Figure 2.
- We implemented *StepLR* to decrease the learning rate every 5 epochs and *CyclicLR* to cycle between $(0.5 \cdot lr, 1.5 \cdot lr)$, but they did not result in any noticeable performance improvement.
- BiDAF + Self-Attention took only 2h:19m to reach EM/F1 score of 64.8/68. QANet took 6h:23m for same number of epochs and more than 8 hours to reach comparable performance.
- Character-level CNN did not improve the performance of either BiDAF or QANet as shown in Table 1.
- *Automatic Mixed Precision (AMP)* brought a 30% reduction of training time for BiDAF but only a 18% reduction of training time for QANet.
- From Figure 3, we can see that both QANet and BiDAF have their relative strengths and weaknesses depending on question type. QANet was an **all-rounder** across all question types while BiDAF **performed poorly on "why" questions** and well on "where" questions. This brought down the performance of the ensemble model in the "why" question category.

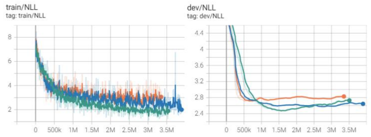

Figure 2. Train and Dev Negative Log-Likelihood Loss (NLL). Green line: BiDAF. Orange line: QANet. Blue line: QANet with Stochastic Depth. We can see that QANet's training loss plateaued after 10-15 epochs.

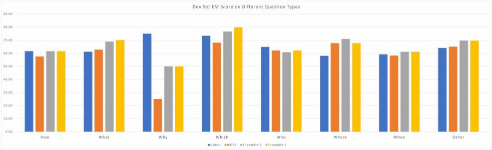

Figure 3. Dev Exact Match comparison based on question category. Blue was QANet. Orange was BiDAF. Gray was Ensemble-5. Yellow was Ensemble-7.

References

[1] Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[3] Adams Wei Yu, David Dohan, Quoc Le, Thang Luong, Rui Zhao, and Kai Chen. Fast and accurate reading comprehension by combining self-attention and convolution. In *International Conference on Learning Representations*, 2018.

https://github.com/jerrylzy/SQuAD-QANet   Stanford CS 224N SQuAD Default Project: Sometimes, Less Is More   jerrylzy@stanford.edu, yukewu@stanford.edu